


LEARNING MADE EASY

C++

Essentials

for
dummies®

A Wiley Brand



Brush up on
core concepts

Jump-start your
skills expansion

Find solutions to
common problems

John Paul Mueller
edited by Ronald Mak

C++ Essentials

^{for}
dummies[®]
A Wiley Brand



C++ Essentials

by John Paul Mueller

EDITED BY **Ronald Mak**

for
dummies[®]
A Wiley Brand

C++ Essentials For Dummies®

Published by: **John Wiley & Sons, Inc.**, 111 River Street, Hoboken, NJ 07030-5774, www.wiley.com

Copyright © 2025 by John Wiley & Sons, Inc. All rights reserved, including rights for text and data mining and training of artificial technologies or similar technologies.

Published simultaneously in Canada

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the Publisher. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at <http://www.wiley.com/go/permissions>.

Trademarks: Wiley, For Dummies, the Dummies Man logo, Dummies.com, Making Everything Easier, and related trade dress are trademarks or registered trademarks of John Wiley & Sons, Inc. and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

LIMIT OF LIABILITY/DISCLAIMER OF WARRANTY: THE PUBLISHER AND THE AUTHOR MAKE NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE ACCURACY OR COMPLETENESS OF THE CONTENTS OF THIS WORK AND SPECIFICALLY DISCLAIM ALL WARRANTIES, INCLUDING WITHOUT LIMITATION WARRANTIES OF FITNESS FOR A PARTICULAR PURPOSE. NO WARRANTY MAY BE CREATED OR EXTENDED BY SALES OR PROMOTIONAL MATERIALS. THE ADVICE AND STRATEGIES CONTAINED HEREIN MAY NOT BE SUITABLE FOR EVERY SITUATION. THIS WORK IS SOLD WITH THE UNDERSTANDING THAT THE PUBLISHER IS NOT ENGAGED IN RENDERING LEGAL, ACCOUNTING, OR OTHER PROFESSIONAL SERVICES. IF PROFESSIONAL ASSISTANCE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL PERSON SHOULD BE SOUGHT. NEITHER THE PUBLISHER NOR THE AUTHOR SHALL BE LIABLE FOR DAMAGES ARISING HEREFROM. THE FACT THAT AN ORGANIZATION OR WEBSITE IS REFERRED TO IN THIS WORK AS A CITATION AND/OR A POTENTIAL SOURCE OF FURTHER INFORMATION DOES NOT MEAN THAT THE AUTHOR OR THE PUBLISHER ENDORSES THE INFORMATION THE ORGANIZATION OR WEBSITE MAY PROVIDE OR RECOMMENDATIONS IT MAY MAKE. FURTHER, READERS SHOULD BE AWARE THAT INTERNET WEBSITES LISTED IN THIS WORK MAY HAVE CHANGED OR DISAPPEARED BETWEEN WHEN THIS WORK WAS WRITTEN AND WHEN IT IS READ.

For general information on our other products and services, please contact our Customer Care Department within the U.S. at 877-762-2974, outside the U.S. at 317-572-3993, or fax 317-572-4002. For technical support, please visit <https://hub.wiley.com/community/support/dummies>.

Wiley publishes in a variety of print and electronic formats and by print-on-demand. Some material included with standard print versions of this book may not be included in e-books or in print-on-demand. If this book refers to media such as a CD or DVD that is not included in the version you purchased, you may download this material at <http://booksupport.wiley.com>. For more information about Wiley products, visit www.wiley.com.

Library of Congress Control Number is available from the publisher.

ISBN 978-1-394-30788-3 (pbk); ISBN 978-1-394-30790-6 (ebk); ISBN 978-1-394-30789-0 (ebk)

Contents at a Glance

Introduction	1
CHAPTER 1: Compiling and Running Your First C++ Application.....	3
CHAPTER 2: Storing Data in C++.....	9
CHAPTER 3: Directing the Application Flow.....	37
CHAPTER 4: Dividing Your Work with Functions.....	61
CHAPTER 5: Splitting Up Source Code Files.....	79
CHAPTER 6: Referring to Your Data through Pointers.....	95
CHAPTER 7: Working with Classes	117
CHAPTER 8: Working with Arrays and Pointers	151
CHAPTER 9: Ten Features for More Advanced C++ Programming	159
Index	171

Table of Contents

INTRODUCTION	1
About This Book	1
Foolish Assumptions	1
Icons Used in This Book	2
Where to Go from Here	2
CHAPTER 1: Compiling and Running Your First C++ Application	3
Looking at a Simple C++ Program	4
Using C++ on Microsoft Windows	4
Using C++ on Apple macOS	5
Using C++ on Linux	6
Working with an Integrated Development Environment	7
CHAPTER 2: Storing Data in C++	9
Storing Values in Variables	9
Creating an integer variable	10
Declaring multiple variables	12
Changing values	13
Setting one variable equal to another	14
Initializing a variable	14
Creating a great name for yourself	15
Manipulating Integer Variables	16
Adding integer variables	16
Subtracting integer variables	19
Multiplying integer variables	20
Dividing integer variables	21
Working with Character Variables	23
Null character	24
Nonprintable and other characters	24
Using Strings	25
Getting a part of a string	26
Changing part of a string	27
Adding onto a string	28
Adding two strings	28
Telling the Truth with Boolean Variables	29
Using Floating-Point Values in a Variable	30

	Printing Float and Double Values	31
	Making Your Code More Readable with Enumerations	33
	Reading from the Console.....	34
CHAPTER 3:	Directing the Application Flow	37
	Filling Your Code with Comments	37
	Evaluating Conditions in C++	39
	Finding the right C++ relational operators.....	40
	Combining multiple conditions	41
	Using If Statements and Conditions	42
	Deciding what if and also what else	43
	Going further with the else and if.....	44
	Making Decisions Using a Conditional Operator.....	45
	Repeating Actions with Statements That Loop.....	47
	Looping for.....	47
	Performing a simple for loop	48
	Going backward	50
	Looping while.....	51
	Doing while	53
	Breaking and Continuing.....	54
	Breaking	54
	Continuing	55
	Nesting Loops.....	56
	Using Switch Statements.....	57
CHAPTER 4:	Dividing Your Work with Functions.....	61
	Dividing Your Work	61
	Calling a Function	63
	Passing a variable	64
	Passing multiple variables	65
	Writing Your Own Functions	66
	Improving On the Basic Function	67
	Using multiple parameters or no parameters	67
	Returning nothing.....	69
	Keeping your variables local.....	70
	Forward references and function prototypes	72
	Writing two versions of the same function.....	74
	Calling All String Functions.....	76
	Inserting a string into a string	76
	Removing parts of a string.....	76
	Replacing parts of a string	77
	Understanding main().....	77

CHAPTER 5:	Splitting Up Source Code Files	79
	Creating Multiple Source Files	79
	Sharing with Header Files	82
	Adding the header only once	85
	Using angle brackets or quotes	85
	Sharing Variables among Source Files	86
	Using the Mysterious Header Wrappers	87
	Using Constants	89
	Understanding Preprocessor Directives	91
	Understanding the basics of preprocessing	92
	Creating constants with <code>#define</code>	93
CHAPTER 6:	Referring to Your Data through Pointers	95
	Considering the Issues with Pointers	95
	Heaping and Stacking the Variables	96
	Getting a variable's address	97
	Changing a variable by using a pointer	98
	Pointing at a string	101
	Pointing to something else	104
	Getting tips on pointer variables	105
	Creating New Pointers	105
	Using <code>new</code>	106
	Using an initializer	108
	Freeing Pointers	108
	Passing Pointer Variables to Functions	111
	Returning Pointer Variables from Functions	115
CHAPTER 7:	Working with Classes	117
	Understanding Objects and Classes	117
	Classifying classes and objects	118
	Describing properties and behaviors	118
	Implementing a class	119
	Separating member function code	119
	Identifying the parts of a class	125
	Working with a Class	126
	Accessing members	126
	Using classes and pointers	128
	Passing objects to functions	130
	Using <code>const</code> parameters in functions	132
	Using the <code>this</code> pointer	133
	Defining standard <code>this</code> pointer usage	134

Starting and Ending with Constructors and Destructors	136
Starting with constructors	136
Ending with destructors	136
Sampling constructors and destructors.....	137
Adding parameters to constructors	138
Building Hierarchies of Classes	141
Overriding member functions.....	143
Generating random numbers	145
Using polymorphism	146
Overloading member functions.....	148
CHAPTER 8: Working with Arrays and Pointers	151
Declaring Arrays	151
Using Arrays and Pointers.....	153
Using Multidimensional Arrays.....	154
Declaring a multidimensional array	155
Considering data type	156
Initializing multidimensional arrays	156
Passing multidimensional arrays	156
Working with Arrays and Command-Line Parameters	157
Allocating an Array on the Heap.....	158
Deleting an Array from the Heap	158
CHAPTER 9: Ten Features for More Advanced C++ Programming.....	159
Conditional Compilation.....	159
Namespaces.....	160
File I/O.....	161
Exception Handling	162
Operator Overloading.....	164
STL Vectors.....	166
STL Maps	166
STL Iterators.....	167
Algorithms Library.....	168
Smart Pointers	169
INDEX.....	171

Introduction

Welcome to *C++ Essentials For Dummies*. This book contains the basic information you need to know to get going with C++ programming. You'll learn how to install and run the compiler, write programs, and use important C++ features like arrays, classes, and pointers to do good object-oriented programming.

C++ is a large and complex language. But whether you're just learning to program or you're an expert in other programming languages, this book contains the essentials of the language for you to quickly become a C++ programmer.

About This Book

C++ Essentials For Dummies contains the information you need to start writing C++ programs. You can compile and run the example programs with the latest C++ 20 version of the compiler, but you can also use the earlier C++ 14 and C++ 17 versions.

This book is not a comprehensive reference for C++. To go beyond what this book covers, your next book should be *C++ All-in-One For Dummies*, 4th Edition by John Paul Mueller, from which this book is derived. However, this book will get you up and running fast with its easy-to-follow *For Dummies* format.

Foolish Assumptions

We made a few basic assumptions about you, the reader:

- » **You own or have access to a relatively modern computer.** The C++ example programs were tested on a Windows computer, but the book shows how to compile and run them just as easily on a Mac or Linux computer.
- » **You're an experienced computer user.** In other words, we assume that you know the basics of using your computer, such as starting programs and working with the file system.

» You're interested in learning how to write programs in the C++ language. That's what this book is about, so it's a fair assumption.

We do *not* assume you have any previous programming experience in C++ or in any other programming language.

Icons Used in This Book

Like any *For Dummies* book, this book is chock-full of helpful icons that draw your attention to items of particular importance. You find the following icons throughout this book:



WARNING

Danger! This icon highlights information that may help you avert disaster.



REMEMBER

Did we tell you about the memory course we took?



TIP

Pay special attention to this icon; it lets you know that some particularly useful tidbit is at hand.

Where to Go from Here

If you're new to programming, you'll benefit most from this book by reading it from start to end. But if you're familiar with programming in another language, especially an object-oriented language like Java, you can pick and choose which chapters and topics to read to learn what's unique about C++.

- » Running a simple C++ program
- » Working with C++ on Microsoft Windows, Apple macOS, and Linux
- » Using an integrated development environment

Chapter 1

Compiling and Running Your First C++ Application

A C++ compiler translates your source files (which contain the text of your application's program code written in C++) into machine code that your computer can understand and run. Different C++ compilers run on the Microsoft Windows, Apple macOS, and Linux operating systems, but they all adhere to the international standard for C++, so if you write your program code in C++, you should be able to compile and run your application on any of the platforms.

This chapter has links to download and install the C++ compiler on the Windows, macOS, and Linux platforms. It demonstrates how to edit, compile, and run a simple C++ application on the command line (that is, in a terminal window) in each of the platforms.

Then it briefly looks at using an integrated development environment (IDE) and shows you the advantages an IDE offers to C++ developers. Many IDEs are available, so this chapter only gives an introduction to using one. The rest of this book assumes you're compiling and running programs on the command line.

Looking at a Simple C++ Program

Use any text editor (such as Windows Notepad, macOS TextEdit, or Linux vi) to create a plain-text file named `hello.cpp` containing the following C++ code:

```
#include <iostream>

using namespace std;

int main()
{
    for (int i = 1; i <= 5; i++)
    {
        cout << i << " Hello, world!" << endl;
    }
}
```

Using C++ on Microsoft Windows

On the Windows platform, the C++ compiler is a utility program called CL that you run on the command line in a special command window.

To install CL, follow these steps:

1. **Go to** <https://visualstudio.microsoft.com/downloads/#build-tools-for-visual-studio-2022>.
2. **Scroll down to Build Tools for Visual Studio and click the Download button.**
An installer downloads onto your computer.
3. **Run the installer to install CL and its tools.**

You must run CL in a special command window. Left-click the Start button at the bottom of the screen. Scroll down to the Visual Studio 2022 folder, and click the down arrow to the right of the folder to reveal the folder's contents. Left-click Developer Command Prompt for VS 2022 to open a command window especially

set up to run CL. On the command line of that window, enter the following command on one line:

```
cl /EHsc /std:c++20 hello.cpp /link /out:hello.exe
```

This command compiles the source file and generates the executable file `hello.exe`. To run the executable file, enter the following command:

```
hello
```

In later chapters, an application may consist of multiple source files. Then, instead of specifying a single source file like `hello.cpp`, you would tell CL to compile all the source files in the directory whose names end with `.cpp`. For example:

```
cl /EHsc /std:c++20 *.cpp /link /out:books.exe
```

These example commands include the `/std:c++20` parameter to specify using the standard 2020 version of C++. You should be able to compile this book's examples with earlier versions of C++ by specifying `/std:c++14` or `/std:c++17`.

The `/link` option says that the next option is meant for the *linkage editor*, a tool that combines the output of the compiler into an executable program. The `/out` parameter tells the linkage editor what to name the executable program.

`/EHsc` tells the compiler how to generate code to handle exceptions, which are runtime errors. Exception handling is briefly covered in Chapter 9.



TIP

For a walk-through on compiling and running a C++ program on the command line, go to <https://learn.microsoft.com/en-us/cpp/build/walkthrough-compiling-a-native-cpp-program-on-the-command-line>. For descriptions of the CL compiler options, go to <https://learn.microsoft.com/en-us/cpp/build/reference/compiler-options-listed-alphabetically>.

Using C++ on Apple macOS

On macOS, you can use the Clang C++ compiler. For instructions on how to install the compiler, go to <https://osxdaily.com/2023/05/02/how-install-gcc-mac>.

After installing the compiler, go to the folder where you stored your source file, `hello.cpp`, and enter the following command in a Terminal window:

```
g++ -std=c++0x -o hello hello.cpp
```

This command runs the C++ compiler to compile the source file and generates the executable file `hello`. To run the executable file, enter the following command:

```
./hello
```

In later chapters, an application may consist of multiple source files. Then, instead of specifying a single source file like `hello.cpp`, you would tell `g++` to compile all the source files in the directory whose names end with `.cpp`. For example:

```
g++ -std=c++20 -o books *.cpp
```

These example commands include the `-std:c++20` parameter to specify using the standard 2020 version of C++. You should be able to compile this book's examples with versions as early as 2011 by specifying `-std=c++11` or `-std=c++0x`. The `-o` parameter gives the name of the executable program.

Using C++ on Linux

On Linux, you can use the GNU C++ compiler. For instructions on how to install the compiler, go to www.cherryservers.com/blog/how-to-install-gcc-on-ubuntu.

After installing the compiler, go to the folder where you stored your source file, `hello.cpp`. Enter the following command in a Terminal window:

```
g++ -std=c++20 -o hello hello.cpp
```

This command runs the C++ compiler to compile the source file and generates the application `hello`. To run the executable file, enter the following command:

```
./hello
```

In later chapters, an application may consist of multiple source files. Then, instead of specifying a single source file, like `hello.cpp`, you would tell `g++` to compile all the source files in the directory whose names end with `.cpp`. For example:

```
g++ -std=c++20 -o books *.cpp
```

These example commands include the `-std=c++20` or `-std=c++2a` parameter to specify using the standard 2020 version of C++. You should be able to compile this book's examples with versions as early as 2011 by specifying `-std=c++11` or `-std=c++0x`). The `-o` parameter gives the name of the executable program.

Working with an Integrated Development Environment

An IDE is an application that combines a syntax-aware code editor, a compiler, a debugger, and often other tools. Many IDEs are cross-platform — they have versions that run on different platforms. Others only run on particular platforms. Some IDEs for C++ include the following:

- » CLion (cross-platform; www.jetbrains.com/clion)
- » Code::Blocks (cross-platform; www.codeblocks.org)
- » Eclipse CDT (cross-platform; <https://projects.eclipse.org/projects/tools.cdt>)
- » Microsoft Visual C++ (<https://visualstudio.microsoft.com/downloads>)
- » Visual Studio (cross-platform; <https://visualstudio.microsoft.com>)
- » Xcode (Apple macOS; <https://developer.apple.com/xcode>)

An IDE's syntax-aware code editor makes it easy to edit C++ code. It can use color-coding to highlight different source code components, such as keywords, and flag syntax errors as you type the code.